**JaJa SDK for Android v1.00 Beta**

**API Description**

Copyright (c) 2012 Silicon Spark pt ltd. All rights reserved. International Patents Pending.

Prepared by: Mercury Development, LLC

Updated: July 3, 2012

**Overview**

The purpose of this document is to provide the high-level description of the JaJa-SDK application programming interface (API). The document includes descriptions of all classes, protocols, properties and methods.

SDK consists of an Eclipse Framework and a sample application that can be used for testing

**SDK Archive content**

|  |  |
| --- | --- |
| JaJa-SDK.framework | JaJa SDK framework binaries |
| JaJa-Sample | JaJa SDK sample source code |
| JaJa-Sample-Bin | JaJa SDK sample build |

**Target environment**

Android 2.3.1 or higher

**Framework usage**

The framework has a simple interface for getting the main parameters. The main control class exported by the Framework is the JajaControlConnection class (see details below).

Framework starts sound detection after calling the start method and

It finishes detection by calling the stop method.

Values can be retrieved from the framework by continuous polling, or using listener methods.

**JajaControlConnection class description**

|  |  |
| --- | --- |
| **Superclass:** | Object |
| **Declared In:** | JajaControlConnection.java |

**Introduction**

JajaControlConnection class provides access to handling control state and receiving

actions from device. JajaControlConnection class can be instantiated several times,

but it will always reference the same private object that is responsible for getting the

device data, but only one can be active at a time.

**Methods**

**start**

Starts the signal receiver.

- void start();

**Exception:**

Throws ConnectionStartedException if another JajaControlConnection instance is already stared

**stop**

Stops the signal receiver.

- (void)stop;

**setJajaControlListener**

Sets the receiver’s listener

void setJajaControlListener(JajaControlListener listener)

**isFirstButtonPressed**

A Boolean value that indicates whether the first button is pressed or not.

boolean isFirstButtonPressed()

**isSecondButtonPressed**

A Boolean value that indicates whether the second button is pressed or not.

boolean isSecondButtonPressed()

**isSignalAvailable**

A Boolean value that indicates whether the signal is available or not. True if a valid value was received from the device at the last moment, false otherwise.

boolean isSignalAvailable()

**getSignalValue**

A double value that represents the pressure level.

signalValue can take any value from 0 to 1. Current JaJa solution handles 1024 discrete pressure levels. Due to compatibility reasons for future versions, these levels are represented as float values from 0 to 1, any other values are invalid;

double getSignalValue()

**isStarted**

A Boolean value that indicates whether the Sound Detector is running or not (this means that the framework is currently listening to the sound and trying to detect the signal received from the device).

boolean isStarted()

**JajaControlListener interface description**

|  |  |
| --- | --- |
| **Extends Interface:** | **NSObject** |
| **Declared In:** | **JajaControlListener.java** |

**Introduction**

JaJa control event listener

**Discussion**

The JajaControlListener defines the listener methods of a JajaControlConnection. Listener methods are called whenever any parameter is changed. Parameters are updated as often as possible without seriously compromising the performance.

Listener methods are also used to notify that the listening has started/stopped and that the device signal is lost / restored.

**Methods**

**firstButtonValueChanged:**

Invoked when the pressure is changed.

void firstButtonValueChanged(boolean isPressed)

**Parameters**

isPressed - The current state of the button.

**jajaControlSignalLost**

Invoked when devive signal is lost.

void jajaControlSignalLost()

**jajaControlSignalRestored**

Invoked when device signal is restored

void jajaControlSignalRestored()

**jajaControlError**

Invoked error occurred during signal processing.

void jajaControlError();

**signalValueChanged**

Invoked when the pressure is changed.

void signalValueChanged(double value)

**Parameters**

value - the current pressure value. 0 - 1

**secondButtonValueChanged:**

Invoked when the pressure is changed.

void secondButtonValueChanged(boolean isPressed)

**Parameters**

isPressed - the current state of the button.